**Dynamic Programming**

Dynamic Programming is a general algorithm design technique for solving problems defined by recurrences with overlapping subproblems. It is invented by American mathematician Richard Bellman in the 1950s to solve Optimization Problems. It is basically a planning, bur named as programming.

Main idea of Dynamic Programming is:

* Set up a recurrence relating a solution to a larger instance to solution of some smaller instances
* Solve smaller instance once
* Record solution in a table
* Extract solution to the initial from that table

Basically Dynamic Programming build up a solution by computing solution to the sub-problems. It doesn’t solve the same sub-problem twice, but rather save the solution so it can be re-used later on. It often used for a large class to optimization problems. Unlike Greedy Algorithms, implicitly it solves all sub-problems.

The process of saving solutions to sub-problems that can be re-used later without redundant computations is called Memoizations. Typically, the solution to sub-problems are saved in a global array, which are later looked up and re-used as needed. Here-

* At each step of computation, first see if the solution to the sub-problem has already been found and saved.
* If so, simply return the solution
* If not, compute the solution, and save it before returning the solution.

Idea of Memoization is to re-use saved solutions, trading off space for time. Any recursive algorithm can be memoized, but only helps if there is redundancy in computing solutions to sub-problems, in other word if there are overlapping sub-problems.

If any recursive algorithm where redundant solutions are computed, Memoization is an appropriate solution. It often called Top-down Dynamic Programming. Motivating the case for Dynamic Programming with Mamoization is a Top-down technique, and then moving on to Dynamic Programming is called a Bottom-up technique.

Dynamic Programming is an algorithm design technique for optimization problems. Like divide and conquer (D&C), Dynamic Programming solves problems by combining solution to sub-problems. Unlike D&C, sub-problems are not independent, they may share sub-sub-problems.

The term Dynamic Programming comes from Control Theory, not computer science. Programming refers to the use of tables(arrays) to construct a solution. In Dynamic Programming we usually reduce time by increasing the amount of space. We solve the problem by solving sub-problems of increasing size and saving each optional solution in a table. The table is then used for finding the optional solution to larger problems. Time is saved since each sub-problem is solved once.

To design a Dynamic Programming Algorithm we have to these steps-

* Characterize the structure of an optimal solution.
* Recursively define the value of an optimal solution.
* Compute the value of an optimal solution in a bottom-up fashion.
* Construct an optimal solution from computed information.